A polynomial-time key recovery attack on Facto-DSA

Simon Abelard[1,2], Ludovic Perret[1,2], and Hao Shi (史豪)[1,3]
[1]Laboratoire de Recherche de l'EPITA (LRE), EPITA, France
[2]Sorbonne Université, CNRS, LIP6, Paris, France
[3]École Polytechnique, Institut Polytechnique de Paris, France
simon.abelard@epita.fr ludovic.perret@epita.fr
hao.shi@polytechnique.edu

July 9, 2026

Abstract

This work introduces a polynomial-time attack on the signature scheme introduced in [GR26] as Facto-DSA. We provide an implementation that breaks all proposed parameter sets, including the largest, in under one minute on a standard laptop. These results question the suitability of multivariate polynomial factorization as a foundation for robust cryptographic schemes.

Keywords: Multivariate Cryptography, Cryptanalysis

1 Introduction

The emergence of quantum computers necessitates adopting cryptographic primitives based on problems that remain computationally hard for quantum adversaries. Among these, multivariate schemes leverage the difficulty of solving systems of polynomial equations, and numerous signature schemes have been constructed on this foundation. In May 2026, the National Institute of Standards and Technology (NIST) advanced nine candidates to the third round of its additional digital-signature competition; four of them – MAYO [Beu21], QR-UOV [FIKT21], SNOVA [WTKC26], and UOV [KPG99] – are multivariate schemes from the UOV family [Nat26].

All of these schemes are variants of the well-established UOV scheme, which has remained secure against cryptanalysis for over 25 years. These schemes employ a trapdoor mechanism based on hidden linear equations, enabling legitimate signers to efficiently generate signatures, while adversaries must solve systems of quadratic equations.

The primary advantage of such schemes is their short signature size, which is beneficial in constrained environments or protocols, such as DNSSEC, where the packet size is limited. However, a significant drawback is the large size of multivariate public keys, although this limitation can be mitigated with static keys.

1.1 Related Work

Multivariate signatures. Most multivariate signature schemes, including round-3 candidates of the ongoing NIST post-quantum signature standardization process, follow the so-called Matsumoto–Imai paradigm introduced in [MI88]. Signing a message requires solving a system of equations. The verifier, on the other hand, only needs to check that the received signature satisfies a set of quadratic equations defined by the sender's public key. While an attacker would need to solve a quadratic system to forge a valid signature, the legitimate signer knows some secrets that allow them to produce a valid signature by solving a linear system.

Factorization-based signatures. In the UOV framework, this secret comes as an isotropic space, yielding public quadratic forms that share a common block of zeros when represented in a secret basis. For a factorization-based signature, e.g. [DMFS26, Gom26], the form of the matrices is even more peculiar in a secret basis. Their diagonal blocks are zero and their off-diagonal blocks are Hankel matrices. For brevity, we do not detail the link between Hankel matrices and factorization; we refer to [DMFS26, Gom26] for a proper explanation.

This paradigm was first proposed in [DMFS26] with the introduction of a signature scheme called X-1 (also referred to as X24), and subsequently expanded in [Gom26] with a scheme called Facto-DSA. This scheme proposes signature and public key sizes of respectively 24 bytes and 12 kilobytes, which are an order of magnitude smaller than standard UOV-type schemes.

Less than a year after its publication, a polynomial-time attack was proposed against X-1 [CLM$^+$26] and the author of [Gom26] realized that it also broke his own scheme. He then proposed countermeasures in [GR26]. Given the short time between his two papers, we shall still use the name Facto-DSA to refer to the updated version including these countermeasures. Although [GR26] indeed thwarts the recent attack from [CLM$^+$26], we show that Facto-DSA can actually be broken using elementary linear algebra.

1.2 Organization of the paper and main results

This work presents an attack that recovers an equivalent secret key in polynomial time. Specifically, given a Facto-DSA public key consisting of n symmetric matrices $Q_1, \ldots, Q_n \in \mathbb{F}_q^{(2n-1)\times(2n-1)}$, the attack constructs n matrices $\hat{G}_i$ of size n that serve as an equivalent private key. To assess the attack's efficiency and substantiate heuristic claims, experimental results are provided along with a software archive to facilitate reproducibility.

The remainder of the paper is organized as follows. Section 2 fixes the notation and describes Facto-DSA; Section 3 presents an overview and de-

tails the three phases of the attack; Section 4 analyzes the computational complexity and reports experimental results.

2 The Facto-DSA signature scheme

Throughout the paper, let $q = p^e$ be an odd prime power, and let $\mathbb{F}_q$ be the finite field with q elements. Recall that a $n \times n$ Hankel matrix is a symmetric matrix of the form

$$H = \begin{pmatrix} h_0 & h_1 & h_2 & \cdots & h_{n-1} \\ h_1 & h_2 & & & h_n \\ h_2 & & & & \vdots \\ \vdots & & & & \vdots \\ h_{n-1} & h_n & h_{n+1} & \cdots & h_{2n-2} \end{pmatrix},$$

so that it can be described by the vector $h = (h_0, \ldots, h_{2n-2}) \in \mathbb{F}_q^{2n-1}$ and we write $H = H(h)$. Let us now briefly describe the Facto-DSA scheme introduced in [GR26].

Key generation. The key generation procedure samples two matrices $S \in \mathbb{F}_q^{2n \times 2n-1}$ and $T \in \mathbb{F}_q^{n \times 2n-1}$, of ranks $2n-1$ and n respectively. Write $E = \mathbb{F}_q^{2n-1}$ and $V = \mathbb{F}_q^{2n}$. For $1 \leq i \leq n$, let t_i^T be the i-th row of T. The vector $t_i \in \mathbb{F}_q^{2n-1}$ defines the Hankel matrix $H_i = H(t_i)$, and we set $M_i = \left(\begin{smallmatrix} 0 & H_i \\ H_i & 0 \end{smallmatrix}\right) \in \mathbb{F}_q^{2n \times 2n}$.

The private key is (S, T), while the public key is made of n symmetric matrices of size $2n-1$ defined by $Q_i = S^T M_i S$. Notice the secret key contains quadratic forms on V while the public key is made of quadratic forms on E.

In [GR26], the author proposes to choose S in $\mathbb{F}_q^{2n \times (2n-c)}$ with c a parameter and call it projected Facto-DSA. This choice is paramount to ensure that the scheme is immune to the attack of [CLM+26], as explained below. In the rest of this paper, Facto-DSA shall refer to the projected Facto-DSA scheme with $c = 1$ as described in this section. We explain in the conclusion why increasing c does not seem like a good option to thwart our attack.

Signature generation. Given a message $m \in \{0,1\}^*$, the signer computes a digest $h = (h_1, \ldots, h_n) \in \mathbb{F}_q^n$. Note that the way to obtain $h \in \mathbb{F}_q^n$ from $m \in \{0,1\}^*$ is not specified in [GR26], but our attack works regardless of how h is built because the process to build h from m is public (otherwise verification would be impossible).

Since T is surjective by construction, the signer then computes a preimage $y \in \mathbb{F}_q^{2n-1}$ of h by T, *i.e.* $Ty = h$, and interprets y as the coefficients of a

polynomial of degree at most $2n-2$. The signer tries to factor this polynomial into a product of two factors a and b of degree smaller than n, and such that the corresponding vector $\begin{pmatrix} a & b \end{pmatrix} \in \mathbb{F}_q^{2n}$ is in $U = \mathrm{Im}(S)$. If any of these conditions fail, the signer retries with another preimage of h.

The signature $\sigma \in E$ is the unique solution of $S\sigma = \begin{pmatrix} \gamma a \\ \gamma^{-1} b \end{pmatrix}$, where $\gamma \neq 0$ is chosen to ensure that the last coordinate of γa and $\gamma^{-1} b$ are equal, which is not always possible as $b_n a_n^{-1}$ may not be a square in $\mathbb{F}_q$. Once again, failure is handled by restarting the process to ensure the signature is unique. Note that the uniqueness of the signature is not mandatory *per se*, it is a design choice in [GR26].

Signature verification. The verification process is as follows: upon receiving a message m' with a signature σ, compute the digest $h' = (h'_1, \ldots, h'_n)$ and, for every $1 \leq i \leq n$, check that $\sigma^T Q_i \sigma = 2h'_i$, where $(Q_1, \ldots Q_n)$ is the sender's public key.

3 Recovering an equivalent secret key

3.1 Overview

As explained in [GR26], Facto-DSA is not safe when S is a square. In that case, the linear system $Q_i R + R^T Q_i = 0$ for $1 \leq i \leq n$ has $R = S^{-1} J S$ as solution, where $J = \begin{pmatrix} I_n & 0 \\ 0 & -I_n \end{pmatrix}$. Once R is obtained, it suffices to solve $SR = JS$ to recover S. Using a rectangular matrix S prevents an attacker from using the relation $Q_i R + R^T Q_i = 0$. Indeed, since $U = \mathrm{Im}\, S$ is no longer stable under J anymore, J will not be a solution.

We show that this strategy can be foiled by finding a space of dimension $2n-2$ which is stable under J. Since $U \subset V$ has dimension $2n-1$, there is a natural decomposition $V = U \oplus \langle g \rangle$, where g is a nonzero element outside of U.

In Section 3.2, we find a vector τ such that $S\tau = Jg$. We can then split $E = \mathcal{W}_\tau \oplus \langle \tau \rangle$ and $V = S(\mathcal{W}_\tau) \oplus \langle S\tau \rangle \oplus \langle g \rangle$. By construction, $\langle S\tau, g \rangle$ is stable under J and so is $S(\mathcal{W}_\tau)$.

In Section 3.3, we restrict the quadratic forms of the public key to $\mathcal{W}_\tau$ so that we can write linear equations of the form $Q_i^{\mathcal{W}_\tau} R + R^T Q_i^{\mathcal{W}_\tau} = 0$. We show that, with overwhelming probability, any solution R has two eigenspaces of dimension $n-1$. This eigenspace decomposition then gives us a basis in which $Q_i^{\mathcal{W}_\tau}$ has the form $Q_i^{\mathcal{W}_\tau} = \begin{pmatrix} 0 & G_i \\ G_i^T & 0 \end{pmatrix}$.

Finally, after computing $(G_1, \ldots G_n)$, we can deduce an equivalent secret key $(\hat{G}_1, \ldots, \hat{G}_n)$ and Section 3.4 shows that it is indeed equivalent, as it allows us to forge any message.

3.2 Finding a stable space of dimension $2n-2$

Consider a linear combination of the Q_i's such that $Q_0 = \sum_{i=1}^{n} c_i Q_i$ is nonsingular, and $M_0 = \sum_{i=1}^{n} c_i M_i$ so that we have $Q_0 = S^T M_0 S$. Because Q_0 has full rank $2n-1$, we must have $\operatorname{rank}(M_0) \geq 2n-1$. Furthermore, the relation $\operatorname{rank}(M_0) = 2\operatorname{rank}(H_0)$ dictates that the rank of M_0 must be even. Thus, we have $\operatorname{rank}(M_0) = 2n$, meaning that M_0 is nonsingular.

Since M_0 is nonsingular, it defines a nondegenerate symmetric bilinear form $B_0(x, y) = x^T M_0 y$ on V. Since $U = \operatorname{Im} S$ is a hyperplane, the orthogonal complement of U with respect to B_0 is a line, denoted by $U^{\perp_0} = \langle g \rangle$. The relation $M_0 J + J M_0 = 0$ implies that Jg lies again in U, as shown in the next lemma.

Lemma 3.1. *Let $U^{\perp_0} = \{w \in V : B_0(w, z) = 0 \text{ for all } z \in U\}$, and let g be a nonzero generator of $U^{\perp_0}$. Then, there exists $\tau \in \mathbb{F}_q^{2n-1}$ such that $S\tau = Jg$.*

Proof. Given that U is a hyperplane and B_0 is nondegenerate, $U^{\perp_0} = \langle g \rangle$ for a nonzero g. Using $M_0 J + J M_0 = 0$, we obtain

$$B_0(g, Jg) = g^T M_0 J g = -g^T J M_0 g = -(Jg)^T M_0 g = -B_0(Jg, g).$$

By the symmetry of B_0, it follows that $B_0(Jg, g) = B_0(g, Jg)$. Over a field with $\operatorname{char}(\mathbb{F}_q) \neq 2$, this gives $B_0(g, Jg) = 0$. Hence, $Jg \in \langle g \rangle^{\perp_0} = U$. Since $U = \operatorname{Im}(S)$, there exists $\tau \in \mathbb{F}_q^{2n-1}$ such that $S\tau = Jg$. □

We now explain how to compute τ from the public matrices Q_i. To this end, for $\lambda = (\lambda_1, \ldots, \lambda_n) \in \mathbb{F}_q^n$, we define $H_\lambda = \sum_{i=1}^{n} \lambda_i H_i$, $M_\lambda = \sum_{i=1}^{n} \lambda_i M_i$, and $Q_\lambda = \sum_{i=1}^{n} \lambda_i Q_i$. Following an idea from the Kipnis–Shamir attack on OV [KS98], we define two endomorphisms of E and V, respectively $\Psi_\lambda = Q_0^{-1} Q_\lambda$ and $\Phi_\lambda = M_0^{-1} M_\lambda$. The goal of this section is to show that τ lies in the eigenspaces of Ψ_λ, which is proven in Proposition 3.1.

Lemma 3.2. *Let $\pi_U : V \to U$ be the projection onto U along $\langle g \rangle$, that is, $w - \pi_U(w) \in \langle g \rangle$. Then $S\Psi_\lambda = \pi_U \Phi_\lambda S$.*

Proof. As $Q_0 = S^T M_0 S$ is nonsingular, the quadratic form B_0 restricted to U is nondegenerate. Consequently, $U \cap U^{\perp_0} = \{0\}$. Given that $g \notin U$, the space V decomposes as $V = U \oplus \langle g \rangle$. Thus, the projection π_U along $\langle g \rangle$ is well-defined.

Let $x, y \in \mathbb{F}_q^{2n-1}$. By the definition of Ψ_λ and Φ_λ, we have

$$B_0(Sx, S\Psi_\lambda y) = x^T Q_0 \Psi_\lambda y = x^T Q_\lambda y = (Sx)^T M_\lambda S y = B_0(Sx, \Phi_\lambda S y).$$

It follows that $B_0(Sx, S\Psi_\lambda y - \Phi_\lambda S y) = 0$ for all x. Since Sx covers all of U, this means

$$S\Psi_\lambda y - \Phi_\lambda S y \in U^{\perp_0} = \langle g \rangle.$$

Furthermore, $S\Psi_\lambda y \in U$. Therefore, $S\Psi_\lambda y$ is exactly the U-component of $\Phi_\lambda Sy$ in the decomposition $V = U \oplus \langle g \rangle$. This proves $S\Psi_\lambda y = \pi_U(\Phi_\lambda Sy)$, for all $y \in \mathbb{F}_q^{2n-1}$. □

Lemma 3.3. *For every integer $k \geq 0$, one has $S\Psi_\lambda^k \tau = \Phi_\lambda^k Jg$ and $\Phi_\lambda^k Jg \in U$. Consequently, for every polynomial $P \in \mathbb{F}_q[X]$, $SP(\Psi_\lambda)\tau = P(\Phi_\lambda)Jg$.*

Proof. We first record the relations used below. The anti-commutation $M_0 J + JM_0 = 0$ and $M_\lambda J + JM_\lambda = 0$ imply

$$J\Phi_\lambda = JM_0^{-1}M_\lambda = -M_0^{-1}JM_\lambda = M_0^{-1}M_\lambda J = \Phi_\lambda J.$$

Moreover, Φ_λ is self-adjoint for B_0, while J is skew-adjoint for B_0:

$$B_0(x, \Phi_\lambda y) = B_0(\Phi_\lambda x, y), \qquad B_0(x, Jy) = -B_0(Jx, y).$$

We now prove that $\Phi_\lambda^k Jg$ belongs to U. Using the relation above, we get

$$B_0(g, \Phi_\lambda^k Jg) = B_0(\Phi_\lambda^k g, Jg) = -B_0(J\Phi_\lambda^k g, g) = -B_0(\Phi_\lambda^k Jg, g).$$

Since B_0 is symmetric, the last term is $-B_0(g, \Phi_\lambda^k Jg)$. As $\mathrm{char}(\mathbb{F}_q) \neq 2$, this gives $B_0(g, \Phi_\lambda^k Jg) = 0$. Since $U = g^{\perp_0}$, we obtain $\Phi_\lambda^k Jg \in U$.

It remains to prove the identity $S\Psi_\lambda^k \tau = \Phi_\lambda^k Jg$. We prove this by induction on k. For $k = 0$, this is just the definition of τ. Assume that $S\Psi_\lambda^k \tau = \Phi_\lambda^k Jg$ for some k. Applying Lemma 3.2 to the vector $\Psi_\lambda^k \tau$, we obtain

$$S\Psi_\lambda^{k+1}\tau = \pi_U\big(\Phi_\lambda S\Psi_\lambda^k \tau\big) = \pi_U\big(\Phi_\lambda^{k+1} Jg\big).$$

By the first part of the proof, $\Phi_\lambda^{k+1} Jg \in U$, so the projection π_U acts trivially on this vector. Hence $S\Psi_\lambda^{k+1}\tau = \Phi_\lambda^{k+1} Jg$.

The induction is complete. By linearity, the same statement holds for any polynomial, which proves the second part of the lemma. □

The above lemmas draw a link between polynomials in Ψ_λ and Φ_λ, which we use below to obtain a relation between their characteristic polynomials.

Proposition 3.1. *Let $C_\lambda = H_0^{-1}H_\lambda$. If χ_{C_λ} is irreducible, then $\chi_{C_\lambda} \mid \chi_{\Psi_\lambda}$ and $\tau \in \ker \chi_{C_\lambda}(\Psi_\lambda)$.*

Proof. By construction, $\Phi_\lambda = M_0^{-1}M_\lambda = \begin{pmatrix} C_\lambda & 0 \\ 0 & C_\lambda \end{pmatrix}$. Hence $\chi_{C_\lambda}(\Phi_\lambda) = 0$ by the Cayley–Hamilton theorem.

Applying Lemma 3.3 to $P = \chi_{C_\lambda}$, we get $S\chi_{C_\lambda}(\Psi_\lambda)\tau = \chi_{C_\lambda}(\Phi_\lambda)Jg = 0$. Since S is injective, $\chi_{C_\lambda}(\Psi_\lambda)\tau = 0$. This proves $\tau \in \ker \chi_{C_\lambda}(\Psi_\lambda)$.

Let us prove that every polynomial f such that $f(\Psi_\lambda)\tau = 0$ is divisible by χ_{C_λ}.

Let $f \in \mathbb{F}_q[X]$ satisfying $f(\Psi_\lambda)\tau = 0$. If $\chi_{C_\lambda} \nmid f$, then, since χ_{C_λ} is irreducible, we have $\gcd(f, \chi_{C_\lambda}) = 1$. Hence there exist $a, b \in \mathbb{F}_q[X]$ such

that $a(X)f(X) + b(X)\chi_{C_\lambda} = 1$. Evaluating this identity with $f = \Psi_\lambda$ and applying it to τ gives

$$\tau = a(\Psi_\lambda)f(\Psi_\lambda)\tau + b(\Psi_\lambda)\chi_{C_\lambda}(\Psi_\lambda)\tau = 0,$$

which contradicts $\tau \neq 0$. Therefore $\chi_{C_\lambda} \mid f$.

Finally, by Cayley–Hamilton, $\chi_{\Psi_\lambda}(\Psi_\lambda) = 0$, so $\chi_{\Psi_\lambda}(\Psi_\lambda)\tau = 0$. Applying the previous paragraph to $f = \chi_{\Psi_\lambda}$ gives $\chi_{C_\lambda} \mid \chi_{\Psi_\lambda}$. □

Remark 3.1. Our attack requires χ_{C_λ} to be irreducible, which is not a condition that we can check without already knowing the secret key. Instead, we factor χ_{Ψ_λ} as it depends only on public information. If χ_{C_λ} is indeed irreducible, then χ_{Ψ_λ} must have an irreducible factor of degree n. The converse may not hold, but our many experiments have never produced a counterexample, leading to the following definition.

Definition 3.1 (Accepted sample)**.** A vector $\lambda \in \mathbb{F}_q^n$ is called an *accepted sample* if the characteristic polynomial χ_{Ψ_λ} of $\Psi_\lambda = Q_0^{-1}Q_\lambda$ has an irreducible factor p_λ of degree n. In that case, we denote $K_\lambda = \ker p_\lambda(\Psi_\lambda)$.

Proposition 3.1 shows that, when χ_{C_λ} is irreducible, we have $p_\lambda = \chi_{C_\lambda}$ and thus $\tau \in K_\lambda$. Therefore, by intersecting sufficiently many spaces K_λ for different accepted samples λ, we expect to obtain the line generated by τ. This yields an algorithm for computing τ, as summarised in the theorem below.

Theorem 3.1 (Correctness of Phase 1)**.** *Upon successful termination, Algorithm 1 returns a nonzero vector τ such that $S\tau = Jg$, up to scalar.*

Proof. By Proposition 3.1, $\tau \in K_\lambda$ for every λ such that χ_{C_λ} is irreducible. Hence, $\tau \in K_\alpha \cap K_\beta$ for any two such α, β.

The algorithm terminates when $\dim L \leq 1$, leaving two options: if $L = \{0\}$, it means that one of the K_λ we considered did not contain τ, which can only happen if λ satisfies Definition 3.1 without χ_{C_λ} being irreducible. Otherwise, the algorithm terminates successfully and returns τ. □

Remark 3.2. We do not prove that our algorithm terminates, although it is to be expected that every intersection reduces the dimension of L. In Section B, we give further arguments as to why $K_\alpha \cap K_\beta$ has dimension 1 in general, as observed in our experiments.

Algorithm 1: Recovering Intersection Vector τ.

Input: Symmetric matrices $Q_1, \ldots, Q_n \in \mathbb{F}_q^{(2n-1)\times(2n-1)}$
Output: Base matrix Q_0 and intersection vector τ
1. Choose base Q_0:

Choose random $c = (c_1, \ldots, c_n) \in \mathbb{F}_q^n$ such that $\det(\sum_i c_i Q_i) \neq 0$
$Q_0 \leftarrow \sum_i c_i Q_i$
2. Recover two K_λ
function RecoverKernel
  while true **do**
    Choose random $\lambda = (\lambda_1, \ldots, \lambda_n) \in \mathbb{F}_q^n$
    $\Psi_\lambda \leftarrow \sum_i \lambda_i Q_0^{-1} Q_i$
    $F_\lambda \leftarrow \chi_{\Psi_\lambda}(X)$
    if F_λ has a unique irreducible factor p_λ of degree n **then**
      $K_\lambda \leftarrow \ker p_\lambda(\Psi_\lambda)$
      return $(\lambda, p_\lambda, K_\lambda)$
    end if
  end while
end function
$(\alpha, p_\alpha, K_\alpha) \leftarrow$ RecoverKernel
$L \leftarrow K_\alpha$
repeat
  $(\beta, p_\beta, K_\beta) \leftarrow$ RecoverKernel
  $L \leftarrow K_\beta \cap L$
until $\dim L \leq 1$

if $L = \{0\}$ **then return** failure
end if
return Q_0 and a nonzero $\tau \in L$

3.3 Recovering an equivalent secret key via eigenspaces

Algorithm 1 returns a nonzero vector $\tau \in E$ such that $S\tau = Jg$, where g spans the B_0-orthogonal of $U = S(E)$. Since Q_0 is nonsingular and $\tau \neq 0$, the row vector $\tau^T Q_0$ is nonzero, so $\mathcal{W}_\tau = \ker(\tau^T Q_0) \subset E$ is a hyperplane of E, *i.e.* $\dim \mathcal{W}_\tau = 2n - 2$. We denote by $Q_i^{\mathcal{W}}$ the quadratic form induced by Q_i on $\mathcal{W}_\tau$. The next lemma shows that $S(\mathcal{W}_\tau) = U \cap J(U)$, which is stable under J. Therefore, J induces an endomorphism R_J on $S(\mathcal{W}_\tau)$ such that for all $1 \leq i \leq n$, we have $Q_i^{\mathcal{W}} R_J + R_J^T Q_i^{\mathcal{W}} = 0$.

To find R_J, we solve the linear system $Q_i^{\mathcal{W}} R + R^T Q_i^{\mathcal{W}} = 0$ and claim that any solution gives us a multiple of R_J (see Lemma 3.6, Lemma 3.7 and Remark 3.4). Let $A = \ker(J - I_{2n})$ and $B = \ker(J + I_{2n})$. Since $V = A \oplus B$, $U \cap J(U)$ can also be expressed as the direct sum of the two eigenspaces of R_J. By computing a basis of each eigenspace, we find a basis in which each Q_i has a block diagonal structure from which we can build an equivalent secret key $(\hat{G}_1, \ldots \hat{G}_n)$.

Lemma 3.4. *We have $S(\mathcal{W}_\tau) = U \cap J(U)$. In particular, $S(\mathcal{W}_\tau)$ is stable under J.*

Proof. Let $x \in E$ and set $u = Sx \in U$. Since $Q_0 = S^T M_0 S$ and $S\tau = Jg$, we have

$$\tau^T Q_0 x = (S\tau)^T M_0 (Sx) = (Jg)^T M_0 u.$$

Using $J^T = J$ and $M_0 J + J M_0 = 0$, this becomes

$$(Jg)^T M_0 u = g^T J M_0 u = -g^T M_0 J u = -B_0(g, Ju).$$

Thus x belongs to $\mathcal{W}_\tau$ exactly when Ju is orthogonal to g with respect to B_0. Since $U^{\perp_0} = \langle g \rangle$ and B_0 is nondegenerate, this is the same as saying that Ju belongs to U.

Therefore the image of $\mathcal{W}_\tau$ under S consists precisely of those vectors $u \in U$ for which $Ju \in U$, that is,

$$S(\mathcal{W}_\tau) = \{u \in U : Ju \in U\}.$$

Since $J^2 = I$, the condition $Ju \in U$ is equivalent to $u \in J(U)$. Hence $S(\mathcal{W}_\tau) = U \cap J(U)$. Finally,

$$J(U \cap J(U)) = J(U) \cap U = U \cap J(U),$$

so this space is stable under J. □

Now that we know that $S(\mathcal{W}_\tau)$ is stable under J, the next lemma shows that both $S(\mathcal{W}_\tau)$ and $\mathcal{W}_\tau$ are direct sums of two $(n-1)$-dimensional subspaces inherited from the decomposition $V = A \oplus B$.

Lemma 3.5. *Let φ be a nonzero linear form on V such that $U = \ker \varphi$. Write $\varphi_A = \varphi|_A$ and $\varphi_B = \varphi|_B$, and assume that $\varphi_A \neq 0$ and $\varphi_B \neq 0$. Then $S(\mathcal{W}_\tau) = \ker \varphi_A \oplus \ker \varphi_B$.*

Consequently, $\mathcal{W}_\tau = W_A \oplus W_B$, where

$$W_A = \{x \in \mathcal{W}_\tau : Sx \in \ker \varphi_A\}, \qquad W_B = \{x \in \mathcal{W}_\tau : Sx \in \ker \varphi_B\}.$$

In particular, $\dim W_A = \dim W_B = n - 1$.

Proof. Let $u = a + b$ with $a \in A$ and $b \in B$. Since $U = \ker \varphi$ and J acts as I_n on A and as $-I_n$ on B, the two conditions $u \in U$ and $Ju \in U$ become

$$\varphi_A(a) + \varphi_B(b) = 0, \qquad \varphi_A(a) - \varphi_B(b) = 0.$$

As $\mathrm{char}(\mathbb{F}_q) \neq 2$, these equations force $\varphi_A(a) = 0$, $\varphi_B(b) = 0$. Conversely, these two conditions imply $u \in U$ and $Ju \in U$. Hence $U \cap J(U) = \ker \varphi_A \oplus \ker \varphi_B$.

Together with Lemma 3.4, this gives

$$S(\mathcal{W}_\tau) = \ker \varphi_A \oplus \ker \varphi_B.$$

Since $\ker \varphi_A \subset A$ and $\ker \varphi_B \subset B$, this is a direct-sum decomposition inside V.

Put it back through the injective map $S : E \to U$ by defining

$$W_A = \{x \in \mathcal{W}_\tau : Sx \in \ker \varphi_A\}, \qquad W_B = \{x \in \mathcal{W}_\tau : Sx \in \ker \varphi_B\}.$$

For every $x \in \mathcal{W}_\tau$, the vector Sx has a unique decomposition $Sx = a_0 + b_0$ with $a_0 \in \ker \varphi_A$ and $b_0 \in \ker \varphi_B$. Because $a_0, b_0 \in S(\mathcal{W}_\tau)$ and S is injective on $\mathcal{W}_\tau$, there are unique $x_A, x_B \in \mathcal{W}_\tau$ such that $Sx_A = a_0$ and $Sx_B = b_0$. Thus $x = x_A + x_B$, with $x_A \in W_A$ and $x_B \in W_B$. Uniqueness follows from the injectivity of S and the fact that $V = A \oplus B$. Therefore $\mathcal{W}_\tau = W_A \oplus W_B$. Finally, φ_A and φ_B are nonzero linear forms on n-dimensional spaces, so $\dim W_A = \dim W_B = n - 1$. □

Remark 3.3. Assuming that φ_A and φ_B behave as random linear forms, the probability that $\varphi_A = 0$ or $\varphi_B = 0$ is $\frac{2(q^n-1)}{q^{2n}-1} \approx 2q^{-n}$ (see Claim 5 of [GR26]). Thus, the assumption on φ_A and φ_B made in Lemma 3.5 holds with overwhelming probability.

We have proven that the eigenspace decomposition of J results in a direct sum decomposition $\mathcal{W}_\tau = W_A \oplus W_B$. We now compute this decomposition explicitly.

Lemma 3.6. *Let $B_{\mathcal{W}} \in \mathbb{F}_q^{(2n-1)\times(2n-2)}$ be a basis of $\mathcal{W}_\tau$, and define $Q_i^{\mathcal{W}} = B_{\mathcal{W}}^T Q_i B_{\mathcal{W}}$. There exists a unique matrix R_J satisfying $SB_{\mathcal{W}} R_J = JSB_{\mathcal{W}}$, which is the representation of J on $U \cap J(U)$ relative to the basis $SB_{\mathcal{W}}$. We have*

$$R_J \in \mathcal{S}_{\mathcal{W}} = \left\{ R \in \mathbb{F}_q^{(2n-2)\times(2n-2)} : Q_i^{\mathcal{W}} R + R^T Q_i^{\mathcal{W}} = 0,\ 1 \le i \le n \right\}.$$

Proof. By Lemma 3.4, the space $S(\mathcal{W}_\tau) = U \cap J(U)$ is stable under J. Therefore, there is a unique endomorphism R_J of $\mathbb{F}_q^{2n-2}$ induced by J through the basis $SB_{\mathcal{W}}$, namely $SB_{\mathcal{W}} R_J = JSB_{\mathcal{W}}$.

Let $x, y \in \mathbb{F}_q^{2n-2}$, and set $u = SB_{\mathcal{W}} x$, $v = SB_{\mathcal{W}} y$. We obtain

$$\begin{aligned} x^T (Q_i^{\mathcal{W}} R_J + R_J^T Q_i^{\mathcal{W}}) y &= x^T (B_{\mathcal{W}}^T Q_i B_{\mathcal{W}} R_J + R_J^T B_{\mathcal{W}}^T Q_i B_{\mathcal{W}}) y \\ &= x^T (B_{\mathcal{W}}^T S^T M_i S B_{\mathcal{W}} R_J + R_J^T B_{\mathcal{W}}^T S^T M_i S B_{\mathcal{W}}) y \\ &= x^T (B_{\mathcal{W}}^T S^T M_i J S B_{\mathcal{W}} + B_{\mathcal{W}}^T S^T J M_i S B_{\mathcal{W}}) y \\ &= u^T (M_i J + J M_i) v = 0. \end{aligned}$$

The above equality holds for all x, y, so we have $Q_i^{\mathcal{W}} R_J + R_J^T Q_i^{\mathcal{W}} = 0$. Since this is true for any $1 \le i \le n$, we have $R_J \in \mathcal{S}_{\mathcal{W}}$.

□

Now that we have shown that R_J satisfies a relation similar to J, we show that its eigenspaces are directly inherited from those of J.

Lemma 3.7. *Assume* $\dim \mathcal{S}_{\mathcal{W}} = 1$. *Let* R *be a nonzero matrix in* $\mathcal{S}_{\mathcal{W}}$. *Then* R *has two eigenspaces of dimension* $n-1$ *whose images by* $B_{\mathcal{W}}$ *are* W_A *and* W_B.

Proof. Since $\dim \mathcal{S}_{\mathcal{W}} = 1$, there exists $\mu \in \mathbb{F}_q^*$, such that $R = \mu R_J$. Let

$$\widetilde{W}_A = \{z \in \mathbb{F}_q^{2n-2} : B_{\mathcal{W}} z \in W_A\}, \qquad \widetilde{W}_B = \{z \in \mathbb{F}_q^{2n-2} : B_{\mathcal{W}} z \in W_B\}.$$

These are the coordinate spaces of W_A and W_B in the basis $B_{\mathcal{W}}$.

By Lemma 3.5, we have $\mathbb{F}_q^{2n-2} = \widetilde{W}_A \oplus \widetilde{W}_B$, and both subspaces have dimension $n-1$. Since J is the identity on A and minus the identity on B, the induced endomorphism has coordinate matrix R_J in the basis $B_{\mathcal{W}}$, and hence R_J is the identity on $\widetilde{W}_A$ and minus the identity on $\widetilde{W}_B$.

Since $\mu \neq 0$ and $\mathrm{char}(\mathbb{F}_q) \neq 2$, the eigenvalues $\pm\mu$ of R are distinct. Therefore $\widetilde{W}_A = \ker(R - \mu I)$ and $\widetilde{W}_B = \ker(R + \mu I)$ (up to swapping the signs). Mapping these two eigenspaces by $B_{\mathcal{W}}$ yields W_A and W_B. □

Once we have computed a decomposition of $\mathcal{W}_\tau$ as a direct sum of two eigenspaces of R, the work of [GR26, Section 2.4] shows that we obtain a basis in which the quadratic forms $Q_i^{\mathcal{W}}$ are represented by matrices of the form $\left(\begin{smallmatrix} 0 & G_i \\ G_i^T & 0 \end{smallmatrix}\right)$. Since $\mathcal{W}_\tau$ has dimension $2n-2$, this means we have a basis in which Q_i can be represented by a $2n-1$ matrix consisting of the above matrix with an additional row and column. The algorithm below computes such a matrix and its correctness is established in Theorem 3.2, building upon the above lemmas.

Remark 3.4. Whenever $\dim \mathcal{S}_{\mathcal{W}} > 1$, Algorithm 2 might fail to recover a multiple of R_J, in which case it aborts and restarts with a different choice for Q_0 and τ. Our experiments showed that this occurs very rarely (see Section 4.3). For brevity, we omit the proof but explain in Section A why this is no surprise.

Algorithm 2: Retrieving an equivalent secret key.

Input: Public Q_i's, Q_0 and τ returned by Algorithm 1
Output: Equivalent trapdoor $(B_A, B_B, e, \hat{G}_i)$ and transformation matrix P
1. Compute the eigenspaces of the endomorphism induced by J
$\mathcal{W}_\tau \leftarrow \ker(\tau^T Q_0)$
Compute a basis $B_{\mathcal{W}} \in \mathbb{F}_q^{(2n-1)\times(2n-2)}$ of $\mathcal{W}_\tau$
for $1 \le i \le n$ **do**
    $Q_i^{\mathcal{W}} \leftarrow B_{\mathcal{W}}^T Q_i B_{\mathcal{W}}$
end for
Compute $\mathcal{S}_{\mathcal{W}} = \left\{ R \in \mathbb{F}_q^{(2n-2)\times(2n-2)} : Q_i^{\mathcal{W}} R + R^T Q_i^{\mathcal{W}} = 0,\ 1 \le i \le n \right\}$

if $\dim \mathcal{S}_{\mathcal{W}} \neq 1$ **then**
    return Failure
end if
Choose a nonzero $R \in \mathcal{S}_{\mathcal{W}}$
Compute the two nonzero eigenvalues μ and $-\mu$ of R
$B_A \leftarrow B_{\mathcal{W}} \times \text{basis}(\ker(R - \mu I)), \quad B_B \leftarrow B_{\mathcal{W}} \times \text{basis}(\ker(R + \mu I))$
2. Compute an equivalent secret key:
Choose $e \notin \text{span}(B_A, B_B)$
$P \leftarrow [B_A \mid B_B \mid e]$
for each i **do**
    $Q_i' \leftarrow P^T Q_i P$
    Write $Q_i' = \begin{pmatrix} 0 & G_i & a_i \\ G_i^T & 0 & b_i \\ a_i^T & b_i^T & c_i \end{pmatrix}$
    $\hat{G}_i \leftarrow \begin{pmatrix} G_i & a_i \\ b_i^T & c_i/2 \end{pmatrix}$
end for
return $\hat{G}_i$ and P

Theorem 3.2 (Correctness of Phase 2)**.** *Let τ be such that $S\tau = Jg$. If Algorithm 2 does not return failure, then it returns a basis $P = [B_A \mid B_B \mid e]$ in which each public matrix Q_i has the following form:*

$$Q_i' = \begin{pmatrix} 0 & G_i & a_i \\ G_i^T & 0 & b_i \\ a_i^T & b_i^T & c_i \end{pmatrix},$$

where $G_i \in \mathbb{F}_q^{(n-1)\times(n-1)}$, $a_i, b_i \in \mathbb{F}_q^{n-1}$, and $c_i \in \mathbb{F}_q$.

Define

$$\hat{G}_i = \begin{pmatrix} G_i & a_i \\ b_i^T & \frac{1}{2}c_i \end{pmatrix}.$$

Then, for every vector written in block coordinates as

$$x = \begin{pmatrix} u \\ v \\ t \end{pmatrix}, \qquad u, v \in \mathbb{F}_q^{n-1}, \quad t \in \mathbb{F}_q,$$

we have

$$x^T Q_i' x = 2 \begin{pmatrix} u \\ t \end{pmatrix}^T \hat{G}_i \begin{pmatrix} v \\ t \end{pmatrix}.$$

Proof. By construction, Phase 2 restricts the quadratic forms Q_i to $\mathcal{W}_\tau$ and compute $R \in \mathcal{S}_{\mathcal{W}}$ as in Lemma 3.6. Since the algorithm succeeded, $\dim \mathcal{S}_{\mathcal{W}} = 1$, R is a multiple of R_J. Hence its two eigenspaces are precisely the two subspaces W_A and W_B of Lemma 3.5.

We first explain why the diagonal blocks vanish. Let R be the recovered operator on $\mathcal{W}_\tau$. If $x, y \in \mathcal{W}_\tau$ lie in the same eigenspace of R, then $Rx = \alpha x$ and $Ry = \alpha y$ for some nonzero scalar α. Since R satisfies

$$Q_i^{\mathcal{W}} R + R^T Q_i^{\mathcal{W}} = 0,$$

we get

$$0 = x^T (Q_i^{\mathcal{W}} R + R^T Q_i^{\mathcal{W}}) y = 2\alpha\, x^T Q_i^{\mathcal{W}} y.$$

Because $\mathrm{char}(\mathbb{F}_q) \neq 2$ and $\alpha \neq 0$, we have $x^T Q_i^{\mathcal{W}} y = 0$. Therefore each restricted form vanishes on $W_A \times W_A$ and on $W_B \times W_B$.

Let B_A and B_B be the bases of W_A and W_B returned by the algorithm, and let e be the extra vector used to complete them to a basis of the public input space. Set

$$P = [B_A \mid B_B \mid e], \qquad B_A^T Q_i B_A = 0, \qquad B_B^T Q_i B_B = 0.$$

Thus

$$Q_i' = P^T Q_i P = \begin{pmatrix} 0 & G_i & a_i \\ G_i^T & 0 & b_i \\ a_i^T & b_i^T & c_i \end{pmatrix},$$

where

$$G_i = B_A^T Q_i B_B, \qquad a_i = B_A^T Q_i e, \qquad b_i = B_B^T Q_i e, \qquad c_i = e^T Q_i e.$$

It remains to rewrite the resulting quadratic form. For $x = (u, v, t)^T$, the block form above gives

$$\begin{aligned} x^T Q_i' x &= 2u^T G_i v + 2u^T a_i t + 2v^T b_i t + c_i t^2 \\ &= 2 \begin{pmatrix} u \\ t \end{pmatrix}^T \begin{pmatrix} G_i & a_i \\ b_i^T & \frac{1}{2} c_i \end{pmatrix} \begin{pmatrix} v \\ t \end{pmatrix}, \end{aligned}$$

which is the desired identity. □

3.4 Forging signature using $\hat{G}_i$

To conclude our attack, let us show that the matrices $\hat{G}_i$ yield an equivalent secret key. To this end, we show that solving the equations $2\hat{u}^T \hat{G}_i \hat{v} = 2h_i$ provides a forgery for $h = (h_1, \ldots, h_n)$. Notice that this yields n equations in $2n-1$ variables. Once we fix $\hat{u}$ to a random value, these equations become linear in $\hat{v}$. Adding a scaling factor γ to ensure that the last coordinates of $\hat{u}$ and $\hat{v}$ are equal yields a linear system of n equations in n unknowns, which is solved by linear algebra in the algorithm.

Algorithm 3: Signature Forgery.

Input: Target hash $h \in \mathbb{F}_q^n$, recovered basis P, bilinear matrices $\hat{G}_1, \ldots, \hat{G}_n$, and matrices $Q_1, \ldots, Q_n$
Output: Forged signature σ
if $h = 0$ **then**
    return $0 \in \mathbb{F}_q^{2n-1}$
end if
Sample $\hat{u} \in \mathbb{F}_q^n$ such that $\hat{u}_n \neq 0$ and $M \leftarrow \begin{pmatrix} \hat{u}^T\hat{G}_1 \\ \vdots \\ \hat{u}^T\hat{G}_n \end{pmatrix}$ is invertible
Solve the linear system $M\hat{v} = h$
if $\hat{v}_n = 0$ **then** restart with new $\hat{u}$
end if
if $\hat{v}_n/\hat{u}_n$ is not a square in $\mathbb{F}_q$ **then** restart with a new $\hat{u}$
end if
Choose $\gamma \in \mathbb{F}_q^*$ with $\gamma^2 = \hat{v}_n/\hat{u}_n$ ($\gamma \neq 0$ since $\hat{v}_n \neq 0$)
$\hat{u}' \leftarrow \gamma\hat{u}, \quad \hat{v}' \leftarrow \gamma^{-1}\hat{v}$
$x' \leftarrow (\hat{u}'_1, \ldots, \hat{u}'_{n-1}, \hat{v}'_1, \ldots, \hat{v}'_{n-1}, \hat{u}'_n)^T$
$\sigma \leftarrow Px'$
return σ

Theorem 3.3 (Forgery using the equivalent secret key)**.** *Assume that Phase 2 returns matrices $\hat{G}_i$ and a basis P as in Theorem 3.2. Let $h \in \mathbb{F}_q^n$ be any target hash. Then Algorithm 3 returns a valid forgery σ for the target hash h.*

Proof. If $h = 0$, the algorithm returns $\sigma = 0$, which satisfies every verification equation. Assume that $h \neq 0$ henceforth. The algorithm samples $\hat{u}$ so that $\hat{u}_n \neq 0$ and there exists a solution $\hat{v}$ to the linear system $\hat{u}^T\hat{G}_i\hat{v} = h_i$ for $1 \leq i \leq n$. Defining M reduces to choosing $M = \begin{pmatrix} \hat{u}^T\hat{G}_1 & \cdots & \hat{u}^T\hat{G}_n \end{pmatrix}^T$ with $\det M \neq 0$. Since $\det M$ is a degree-n polynomial in the $\hat{u}_i$, the Schwartz-Zippel lemma implies that $\Pr[\det M = 0] \leq n/q$. For our parameter sets, where $q = 251$ and $n \in \{25, 48, 94\}$, the worst-case failure probability is at most $94/251 \approx 37.5\%$. Thus, a valid invertible matrix M is obtained with probability at least 62.5%, so a proper vector $\hat{u}$ is typically found in a few random trials.

Once a proper $\hat{u}$ is sampled, we compute $\hat{v}$ by solving the linear equation $M\hat{v} = h$. If $\hat{v}_n = 0$, or if $\hat{v}_n/\hat{u}_n$ is not a square in $\mathbb{F}_q$, we sample another $\hat{u}$ and restart. The first event should occur with probability $1/q$ and the second with probability roughly $1/2$.

Let us now set $\hat{u}' = \gamma\hat{u}$ and $\hat{v}' = \gamma^{-1}\hat{v}$, so that we still have $(\hat{u}')^T\hat{G}_i\hat{v}' = \hat{u}^T\hat{G}_i\hat{v} = h_i$. This rescaling is done to ensure that $\hat{u}'_n = \gamma\hat{u}_n = \gamma^{-1}\hat{v}_n = \hat{v}'_n$, which is required for the signature $x' = (\hat{u}'_1, \ldots, \hat{u}'_{n-1}, \hat{v}'_1, \ldots, \hat{v}'_{n-1}, \hat{u}'_n)^T$ to be valid.

Let $\sigma = Px'$ and i be any integer in $\{1, \ldots, n\}$. By Theorem 3.2, we have $\sigma^T Q_i \sigma = (x')^T P^T Q_i P x' = 2(\hat{u}')^T\hat{G}_i\hat{v}' = 2h_i$. Therefore, σ is a valid

signature for h. □

4 Complexity and Experiments

4.1 Complexity

We count field operations over $\mathbb{F}_q$, so that it suffices to multiply our estimates by $\log q$ to obtain a complexity in bit operations. Let $r = 2n-1$, $d = 2n-2$ and let ω denote a feasible exponent for matrix multiplication.

Our complexity analysis is based on the description we presented in Section 3, but our implementation uses a trick to speed up the computation of R, as described in Section 4.2.

Phase 1. The first tasks in Algorithm 1 are to compute all the $Q_0^{-1}Q_i$, then each iteration of RecoverKernel computes and factors χ_{Ψ_λ}. This respectively costs $O(nr^\omega)$ and $O(nr^2 + r^\omega + r^3 \log q)$ field operations, using $O(r^3 \log q)$ as a conservative bound for factoring a degree-r polynomial over $\mathbb{F}_q$. For each λ, computing $K_\lambda = \ker p_\lambda(\Psi_\lambda)$ costs at most $O(nr^\omega)$ with dense matrix arithmetic.

Intersecting two recovered kernels can be done by the standard block-kernel method: if B_α and B_β are column basis matrices for K_α and K_β, then $K_\alpha \cap K_\beta$ is spanned by the vectors $B_\alpha x = B_\beta y$ obtained from $(x, y) \in \ker[\, B_\alpha \mid -B_\beta \,]$. One kernel computation on an $r \times O(r)$ matrix suffices, so the cost is $O(r^\omega)$ field operations.

When some conditions are not met, Algorithm 1 must restart with a different sample. Assuming that χ_{C_λ} behaves as a random polynomial, the probability of it being irreducible is $p_{irr}(n, q) = 1/n + O(q^{-\frac{n}{2}})$. This is a quite well-known result which can be found in [GP97], for instance. The number of required λ to recover τ is expected to be constant (in practice, 2 of them are enough). The probability that Q_0 is not invertible is in $o(1)$. Therefore, using the fact that $r = O(n)$, Algorithm 1 requires $O\big(n^{\omega+2} + n^4 \log q\big)$ arithmetic operations.

Remark 4.1. In practice, only a few tries were enough to obtain an irreducible χ_{C_λ} instead of $O(n)$. Since our probabilistic bound already suffices for a polynomial-time attack, we did not investigate further.

Phase 2. Algorithm 2 first restricts the public matrices to $\mathcal{W}_\tau$ by computing $Q_i^{\mathcal{W}} = B_{\mathcal{W}}^T Q_i B_{\mathcal{W}}$. This costs $O\big(n(r^2 d + r d^2)\big) = O(n^4)$ field operations, since both r and d are in $O(n)$. The dominant step is solving $Q_i^{\mathcal{W}} R + R^T Q_i^{\mathcal{W}} = 0$ for $1 \le i \le n$. This linear system has $O(n^2)$ variables and $O(n^3)$ equations. Solving the resulting overdetermined rectangular system by classical Gaussian elimination costs $O(n^7)$ (or $O(n^{2\omega+1})$ using fast

linear algebra). The subsequent eigenspace computation and block-form extraction are polynomial in d and are dominated by the above cost. Therefore, Algorithm 2 runs in $O(n^7)$ field operations.

Phase 3. In Algorithm 3, each forgery trial builds an $n \times n$ matrix M whose i-th row is $\hat{u}^T \hat{G}_i$ and solves $M\hat{v} = h$. Solving this system requires $O(n^3)$ field operations, which dominate the cost of the other steps. As discussed in the proof of Theorem 3.3, the expected number of trials required for Algorithm 3 to terminate is bounded by a constant; it runs in expected time $O(n^3)$ field operations.

Total Complexity. According to the above analysis, our attack runs in expected time $O\big(n^4 \log q + n^7\big)$ which is indeed polynomial in all parameters of Facto-DSA. In a cryptographic setting, n has much more impact on security than $\log q$ so it is reasonable to further simplify the cost of our attack to $O(n^7)$ field operations. As mentioned above, we decided to optimize our attack to speed up our experiments. Therefore, the algorithm we implemented has complexity $O(n^5)$ as mentioned in Section 4.2.

4.2 Reducing the Number of Unknowns in Phase 2

The bottleneck of our attack is the computation of R via solving $Q_i^{\mathcal{W}} R + R^T Q_i^{\mathcal{W}} = 0$. These equations have $(2n-2)^2$ unknowns but our implementation uses a technique to make the number of unknowns linear rather than quadratic.

Sparse construction of $\mathcal{W}_\tau$. The hyperplane $\mathcal{W}_\tau$ is defined by the public linear equation $\ell x = 0$, where $\ell = \tau^T Q_0$. Instead of computing a dense kernel basis, we explicitly construct a sparse one. Choosing a pivot index p such that $\ell_p \neq 0$, we set $c_i = \ell_i / \ell_p$ and define $b_i = e_i - c_i e_p$ for every $i \neq p$, where e_k denotes the k-th standard basis vector of $\mathbb{F}_q^{2n-1}$. Since $\ell b_i = 0$, the vectors b_i for $i \neq p$ naturally form a basis of $\mathcal{W}_\tau$, with each basis vector containing at most two nonzero entries.

Let $B_{\mathcal{W}}$ be the matrix whose columns are b_i. For any public matrix Q, its restriction is $Q^{\mathcal{W}} = B_{\mathcal{W}}^T Q B_{\mathcal{W}}$. The entries are computed directly as

$$Q_{i,j}^{\mathcal{W}} = b_i^T Q b_j = (e_i - c_i e_p)^T Q (e_j - c_j e_p) = Q_{i,j} - c_j Q_{i,p} - c_i Q_{p,j} + c_i c_j Q_{p,p},$$

where $i, j \neq p$. The implementation can therefore restrict the public matrices to $\mathcal{W}_\tau$ without forming dense intermediate products, saving time and memory; this is particularly important for the largest parameter set, where $\dim \mathcal{W}_\tau = 2n - 2 = 186$.

Reducing the number of unknowns. Set $d = n - 1$, and let $A_i \in \mathrm{Sym}_{2d}(\mathbb{F}_q)$ be the restriction of the public matrices to $\mathcal{W}_\tau$. The endomorphism R must satisfy $A_i R + R^T A_i = 0$ for all i. Solving this directly requires $(2d)^2$ unknowns. We aggressively reduce this dimension by searching R within a smaller space.

Choose two random linear combinations A_0, A_1 of the restricted matrices such that A_0 is invertible, and define $C = A_0^{-1} A_1$. If R satisfies $A_0 R + R^T A_0 = 0$, then $R^T A_0 = -A_0 R$. Since $A_1 = A_0 C$, applying the constraint to A_1 yields

$$0 = A_1 R + R^T A_1 = A_0 C R + R^T A_0 C = A_0 C R - A_0 R C = A_0 (C R - R C).$$

Since A_0 is invertible, $CR = RC$. Conversely, $CR = RC$ together with $A_0 R + R^T A_0 = 0$ implies the equation for A_1. Thus every solution R lies in the centralizer of C, $\mathrm{Cent}(C) = \{X \in Mat_{2d}(\mathbb{F}_q) : XC = CX\}$.

In general, a properly sampled pair A_0, A_1 yields an endomorphism C composed of two cyclic blocks of dimension d. We can verify this property efficiently: the pair is accepted if there exists a degree-d polynomial f such that $\chi_C = f^2$ and the minimal polynomial of C is exactly f.

To exploit this structure, we find two vectors v_1, v_2 whose corresponding Krylov matrices $K(v) = \big[v,\ Cv,\ \ldots,\ C^{d-1}v\big]$ satisfy

$$\operatorname{rank} K(v_1) = \operatorname{rank} K(v_2) = d, \qquad \operatorname{rank}\big[K(v_1) \mid K(v_2)\big] = 2d.$$

The columns of $K(v_1)$ and $K(v_2)$ provide a basis in which C takes the block-diagonal form $C = \mathrm{diag}(D, D)$, where D is a cyclic $d \times d$ matrix. Any matrix R commuting with C can be written in block form as

$$R = \begin{pmatrix} R_{11} & R_{12} \\ R_{21} & R_{22} \end{pmatrix}, \qquad R_{ij} \in Mat_d(\mathbb{F}_q).$$

The condition $CR = RC$ is equivalent to $R_{ij} D = D R_{ij}$ for all $1 \le i, j \le 2$. Since D is cyclic, any matrix commuting with it must be a polynomial in D. Thus, each block takes the form $R_{ij} = h_{ij}(D)$ with $\deg h_{ij} < d$, leaving only d free coefficients per block.

Consequently, $\mathrm{Cent}(C) \simeq M_2(\mathbb{F}_q[D])$, a space of dimension merely $4d = 4(n-1)$. The implementation parametrizes R as

$$R = \begin{pmatrix} a(D) & b(D) \\ c(D) & e(D) \end{pmatrix}, \qquad a, b, c, e \in \mathbb{F}_q[t]_{<d},$$

and imposes the linear constraints $A_i R + R^T A_i = 0$ exclusively on these $4(n-1)$ polynomial coefficients.

Incremental filtering and validation. The centralizer step does not directly output R. Instead, it constructs a smaller space $\mathcal{C}_0 \subseteq \mathbb{F}_q^{(2n-2)\times(2n-2)}$ which is expected to contain a multiple of R_J. Assuming that this is indeed the case, searching R_J within $\mathcal{C}_0$ amounts to reducing the dimension of the search space from $(2n-2)^2$ to merely $4(n-1)$.

To isolate a multiple of R_J, we check the equations one by one. Starting from $\mathcal{C}_0$, we define a descending sequence of candidate spaces:

$$\mathcal{C}_j = \left\{R \in \mathcal{C}_{j-1} : Q_j^{\mathcal{W}} R + R^T Q_j^{\mathcal{W}} = 0\right\}, \qquad j = 1, 2, \ldots, n.$$

At every step, the number of equations to satisfy increases and the number of candidates for R decreases. Once the dimension of $\mathcal{C}_j$ drops to one for some index j, the unique nonzero matrix R (up to scalar) is returned. To ensure correctness, this matrix is checked against the full set of restricted equations:

$$Q_i^{\mathcal{W}} R + R^T Q_i^{\mathcal{W}} = 0, \qquad 1 \le i \le n.$$

If this final validation fails, the candidate space is discarded and the process is restarted, resulting in a Las Vegas algorithm.

4.3 Benchmarks

Implementation. The attack was implemented in SageMath. The implementation of the scheme in [GR26] was used to generate Facto-DSA public keys $Q_1, \ldots, Q_n \in \mathrm{Sym}(E)$ at random. These matrices were the only input used by our attack.

Parameter sets. We benchmark our attacks on the three projected Facto-DSA parameter sets proposed in [GR26, Table 1]. They use $q = 251$, $m = n$, $r = 2n - 1$, and $c = 1$, with $n \in \{25, 48, 94\}$. These correspond respectively to the parameter labels Facto-DSA-128, Facto-DSA-256, and Facto-DSA-512. In terms of NIST security levels, the first two correspond to levels I and V while the last one is much stronger and allegedly offers 512 bits of security.

Experimental setup. For each of these three parameter sets, we generated 20 independent private/public key pairs. We then run the attack of Section 3 with the trick described in Section 4.2. To check that the output matrix is indeed an equivalent secret key, we sampled 5 independent random targets $h \in \mathbb{F}_q^n$, ran the forgery procedure of Algorithm 3, and verified that it outputs valid signatures.

Environment. All experiments were run sequentially with one worker on `Ubuntu 24.04.4 LTS` x86_64, on an AMD Ryzen 9 5900HX with 32 GiB of memory, using `SageMath 10.8` and `Python 3.12.13`.

Code is available at `https://github.com/MingLLuo/facto-dsa-attack`.

Table 1: Average benchmark results over 20 public keys. For each key, the attack reconstructs an equivalent secret key and uses it to sign 5 random targets. Phase 3 reports the total time needed to forge all 5 targets.

Instance	n	r	Phase 1 (s)	Phase 2 (s)	Phase 3 (s)	Total (s)
Facto-DSA-128	25	49	0.10	0.31	0.01	0.42
Facto-DSA-256	48	95	0.86	2.23	0.01	3.10
Facto-DSA-512	94	187	9.43	17.64	0.06	27.13

Validation of our heuristics. All benchmark instances completed successfully. In every run, the implementation verified that our heuristics held. In particular, while running Algorithm 1, we always verified that the intersection of only two spaces $K_\alpha \cap K_\beta$ had dimension 1 and was therefore sufficient to recover τ.

While running Algorithm 2, we always confirmed that $\dim \mathcal{S}_{\mathcal{W}} = 1$, meaning that the matrix R we compute is necessarily a multiple of R_J. We also checked that W_A and W_B both had dimension $n-1$ and were totally isotropic for all public quadratic forms Q_i, thus ensuring the block structure of the matrices Q'_i.

Effect of the centralizer trick. The technique described in Section 4.2 reduces the number of unknowns in the linear systems used to find R from $(2n-2)^2$ to $4(n-1)$. For the three candidate parameter sets, the reductions are respectively from 2304 to 96 variables, from 8836 to 188 variables, and from 34596 to 372 variables. These reductions explain why our implementation of Algorithm 2 remains practical even for the largest tested instance.

5 Conclusion

We have proposed a polynomial-time key recovery attack against Facto-DSA, along with code that breaks instances claimed to be secure at 512 bits. Our attack is conceptually simpler than the one on X-1 described in [CLM+26], so it would be interesting to see whether it can be adapted to target X-1 as well.

Regarding Facto-DSA, we wonder whether increasing the corank of S, denoted by c in [GR26], could thwart our attack. We expect that it would require us to find a space of dimension c instead of finding τ, and the matrices Q'_i would contain a smaller block G_i, requiring c additional rows and columns. Increasing the corank could be addressed via an exhaustive search, incurring $O(q^{c-1})$ additional operations.

Conversely, increasing c would render the signing process impractical. As noted in [GR26, Section 3.3], signing requires identifying u and v in $\mathbb{F}_q^c$ such that $u = \gamma v$ for some square γ in $\mathbb{F}_q$. While this condition is met

approximately half the time for $c = 1$, the probability of success decreases to $O(q^{1-c})$ for $c > 1$. Consequently, the signing time would increase by a factor of $O(q^{c-1})$, making the scheme infeasible for larger c.

References

[Beu21] Ward Beullens. Mayo: practical post-quantum signatures from oil-and-vinegar maps. In *International Conference on Selected Areas in Cryptography*, pages 355–376. Springer, 2021.

[CLM+26] Alexandre Camelin, Thai Hung Le, Brice Minaud, Phong Q. Nguyen, and Florian Tousnakhoff. X24 down: Cryptanalysis of hankel-based multivariate signatures. Cryptology ePrint Archive, Paper 2026/803, 2026.

[DMFS26] Irene Di Muzio, Martin Feussner, and Igor Semaev. Multivariate signatures with polynomial factorization. In *International Conference on Post-Quantum Cryptography*, pages 355–369. Springer, 2026.

[FIKT21] Hiroki Furue, Yasuhiko Ikematsu, Yutaro Kiyomura, and Tsuyoshi Takagi. A new variant of unbalanced oil and vinegar using quotient ring: QR-UOV. In *International Conference on the Theory and Application of Cryptology and Information Security*, pages 187–217. Springer, 2021.

[Gom26] Borja Gomez. Factorisation-based multivariate schemes: Structural properties and new constructions. *Cryptology ePrint Archive*, 2026.

[GP97] Shuhong Gao and Daniel Panario. Tests and constructions of irreducible polynomials over finite fields. In *Foundations of Computational Mathematics: Selected Papers of a Conference Held at Rio de Janeiro, January 1997*, pages 346–361. Springer, 1997.

[GR26] Borja Gómez Rodríguez. Reduction attacks and modifier effects in factorisation-based multivariate signatures. ResearchGate preprint, 05 2026.

[KPG99] Aviad Kipnis, Jacques Patarin, and Louis Goubin. Unbalanced oil and vinegar signature schemes. In *International Conference on the Theory and Applications of Cryptographic Techniques*, pages 206–222. Springer, 1999.

[KS98] Aviad Kipnis and Adi Shamir. Cryptanalysis of the oil and vinegar signature scheme. In *Annual international cryptology conference*, pages 257–266. Springer, 1998.

[MI88] Tsutomu Matsumoto and Hideki Imai. Public quadratic polynomial-tuples for efficient signature-verification and message-encryption. In *Workshop on the Theory and Application of of Cryptographic Techniques*, pages 419–453. Springer, 1988.

[Nat26] National Institute of Standards and Technology. Nine candidates advance to the third round of the additional digital signatures for the PQC standardization process, May 2026.

[WTKC26] Lih-Chung Wang, Po-En Tseng, Yen-Liang Kuan, and Chun-Yen Chou. A simple noncommutative UOV scheme. *Taiwanese Journal of Mathematics*, 30(3):445–479, 2026.

A The solution set of $Q_iR + R^TQ_i = 0$

We explain why the solution space

$$\mathcal{S}_{\mathcal{W}} = \left\{ R \in \mathbb{F}_q^{(2n-2)\times(2n-2)} : Q_i^{\mathcal{W}} R + R^T Q_i^{\mathcal{W}} = 0,\ 1 \le i \le n \right\}$$

is expected to be one-dimensional.

Recall that we have computed a direct sum decomposition $\mathcal{W}_\tau = W_A \oplus W_B$. Let $d = n - 1$. Choosing a basis of $\mathcal{W}_\tau$ adapted to this decomposition, the restriction of the public matrices Q_i, denoted by $\widetilde{Q}_i$, takes the form

$$\widetilde{Q}_i = \begin{pmatrix} 0 & G_i \\ G_i^T & 0 \end{pmatrix}, \qquad G_i \in \mathbb{F}_q^{d\times d}.$$

Let us now write $R = \left(\begin{smallmatrix} X & Y \\ Z & W \end{smallmatrix}\right)$, with blocks in $\mathbb{F}_q^{d\times d}$. Substituting this into the equation $\widetilde{Q}_i R + R^T \widetilde{Q}_i = 0$, we get

$$\begin{pmatrix} G_iZ + Z^TG_i^T & G_iW + X^TG_i \\ G_i^TX + W^TG_i^T & G_i^TY + Y^TG_i \end{pmatrix} = 0.$$

1. **Diagonal blocks:** $G_iZ + (G_iZ)^T = 0$ and $G_i^TY + (G_i^TY)^T = 0$. Hence, G_iZ and G_i^TY must be skew-symmetric matrices for all $1 \le i \le n$. Given that the n independent generic matrices G_i are derived from random Hankel combinations, this overdetermined system will generally only have the trivial solution $Z = Y = 0$.

2. **Off-diagonal blocks:** $G_iW + X^TG_i = 0$. Assuming G_1 is one invertible block among the G_i's, we can write $X^T = -G_1WG_1^{-1}$. Substituting this into the equation for an arbitrary G_i yields $G_iW = G_1WG_1^{-1}G_i$, which rearranges to

   $$(G_1^{-1}G_i)W = W(G_1^{-1}G_i).$$

   Hence, W lies in the centralizer of the algebra generated by the matrices $\{G_1^{-1}G_i\}_{i=1}^n$. In general, this centralizer is $\mathbb{F}_q \cdot I_d$, so $W = \mu I_d$, which in turn gives $X^T = -\mu I_d$.

Consequently, any solution $R \in \mathcal{S}_{\mathcal{W}}$ must either be a scalar multiple of $R_J = \left(\begin{smallmatrix} -I_d & 0 \\ 0 & I_d \end{smallmatrix}\right)$ or satisfy the above two conditions. Finding a matrix that satisfies the first condition is already unlikely, so the solution space is, with very high probability, limited to multiples of R_J.

B Two accepted samples are enough to recover τ

The first phase of the attack intersects kernels of the form $K_\lambda = \ker p_\lambda(\Psi_\lambda)$. By Proposition 3.1, every accepted sample contains the hidden line $\langle\tau\rangle$ in K_λ. The following lemma explains why, under the random subspace model, two such kernels are already enough to isolate this line with high probability.

Lemma B.1. *Let α and β be two accepted samples such that χ_{C_α} and χ_{C_β} are irreducible and assume $\dim K_\alpha = \dim K_\beta = n$. Since $\langle\tau\rangle \subseteq K_\alpha \cap K_\beta$ by Proposition 3.1, write $E = \overline{E} \oplus \langle\tau\rangle$, $K_\alpha = \overline{K_\alpha} \oplus \langle\tau\rangle$ and $K_\beta = \overline{K_\beta} \oplus \langle\tau\rangle$. Then $\dim \overline{E} = 2n-2$ and $\dim \overline{K_\alpha} = \dim \overline{K_\beta} = n-1$. If we fix $\overline{K_\alpha}$ and assume $\overline{K_\beta}$ is distributed uniformly at random among all $(n-1)$-dimensional subspaces of $\overline{E}$, we have*

$$\Pr[K_\alpha \cap K_\beta \neq \langle\tau\rangle] \leq \frac{1}{q-1}.$$

Proof. Note that $K_\alpha \cap K_\beta = \langle\tau\rangle$ if and only if $\overline{K_\alpha} \cap \overline{K_\beta} = \{0\}$. Since $\dim \overline{K_\alpha} + \dim \overline{K_\beta} = \dim \overline{E}$, this intersection is equivalent to $\overline{K_\alpha} \oplus \overline{K_\beta} = \overline{E}$.

Fix an $(n-1)$-dimensional subspace $\overline{K_\alpha}$ in $\overline{E}$ and let $W \cong \overline{E}/\overline{K_\alpha}$ be the complement of subspace $\overline{K_\alpha}$. Any $(n-1)$-dimensional subspace $\overline{K_\beta}$ that satisfies $\overline{K_\alpha} \cap \overline{K_\beta} = \{0\}$ can be represented as $\overline{K_\beta} = \{w + f(w) : w \in W\}$, where $f : W \to \overline{K_\alpha}$ is a linear map.

Since $\dim \overline{K}_\alpha = \dim W = n-1$, the number of such linear maps is $q^{(n-1)^2}$. The total number of $(n-1)$-dimensional subspaces in $\overline{E}$ is given by the Gaussian binomial coefficient $\binom{2n-2}{n-1}_q$. Therefore,

$$\begin{aligned}\Pr[\overline{K_\alpha} \cap \overline{K_\beta} = \{0\}] &= \frac{q^{(n-1)^2}}{\binom{2n-2}{n-1}_q} = \prod_{j=1}^{n-1} \frac{1-q^{-j}}{1-q^{-(n-1+j)}} \\ &\geq \prod_{j=1}^{n-1}(1-q^{-j}) \geq 1 - \sum_{j=1}^{n-1} q^{-j} \geq 1 - \frac{1}{q-1}.\end{aligned}$$

Hence, $\Pr[K_\alpha \cap K_\beta = \langle\tau\rangle] \geq 1 - \frac{1}{q-1}$, which means $\Pr[K_\alpha \cap K_\beta \neq \langle\tau\rangle] \leq \frac{1}{q-1}$. □